International Journal of Production Economics 199 (2018) 193–198

Contents lists available at ScienceDirect

International Journal of Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Polynomial algorithm of inventory model with complete backordering and correlated demand caused by cross-selling

Ren-Qian Zhang [a], Meng Yi [a], Qi-Qi Wang [a], Chen Xiang [a,b,*]

[a] *School of Economics and Management, Beihang University, Beijing 100191, China*
[b] *David D. Reh School of Business, Clarkson University, Potsdam, NY, USA*

ARTICLE INFO

Keywords:
Inventory
Joint replenishment problem
Polynomial algorithm
Complete backordering
Cross-selling

ABSTRACT

In a paper published in the International Journal of Production Economics (IJPE) [Zhang, R., Kaku, I., Xiao, Y., 2012. Model and heuristic algorithm of the joint replenishment problem with complete backordering and correlated demand. *International Journal of Production Economics* 139 (1), 33–41], the authors proposed a joint replenishment problem (JRP) model with complete backordering and correlated demand caused by cross-selling. The model was transformed into minimizing a function with respect to multiples of a major item's order cycle, and a heuristic algorithm was developed for near-optimal solutions. In this paper, we reinvestigate the problem and analyze the mathematical property of the model to develop an exact algorithm. The algorithm can obtain global optima and exhibits polynomial complexity.

1. Introduction

The multi-item inventory management problem has been studied for several decades, for which the joint replenishment problem (JRP) is the most representative topic (Goyal, 1974; Khouja and Goyal, 2008). The classic JRP supposes that the demand for items is deterministic and inventory replenishments are related to one another due to sharing of the common/major ordering cost. In recent years, the JRP model has been further extended to deal with transportation costs (Venkatachalam and Narayanan, 2016), stochastic demand (Braglia et al., 2016a; Lee and Lee, 2018), and perishable or deteriorating items (Kouki et al., 2016; Ai et al., 2017), among others. Certain extensions simultaneously consider multiple factors, including stochastic demand, the backorders-lost sales mixture, controllable lead time, and changeable ordering costs (Braglia et al., 2016b, 2017). Cunha et al. (2017) proposed a model for the multi-item economic lot-sizing problem, which extends JRP to remanufacturing contexts.

It is noticeable that the above work related to the multi-item inventory/production problem holds an implicit assumption: the item has no externalities, which means that the demand for a given item does not affect that for any other items. However, item demands are frequently interrelated in numerous economic systems due to the externalities of products and consumption (Turnovsky and Monteiro, 2007; Hashimoto and Matsubayashi, 2014). In recent decades, commodity externalities have been introduced into inventory models, which can be classified as two types: positive and negative (Netessine and Zhang, 2005). For inventory management, negative externality frequently emerges as substitution among items (Parlar, 1988; Lippman and McCardle, 1997; Netessine and Rudi, 2003; Zhao and Atkins, 2008; Huang et al., 2011), while positive externality, often emerging as a type of association between items, can be caused by cross-selling (Zhang et al., 2012, 2014).

Cross-selling implies that the demand for or sale of an item will lead to an additional demand for its associated items; conversely, if an item is stocked out, the demand for its associated items will decrease. Considering this phenomenon, Zhang et al. (2012) assumed that the sale of a major item affects the demands for multiple minor items. That is, the demand rate of the minor associated items will be increased due to cross-selling with the sale of the major item, but decreased to a certain extent when the major item is stocked out, resulting in an absence of cross-selling. In their model, all stocked out demand for the major item is completely backordered. Based on the above assumption, Zhang et al. (2012) proposed a JRP model considering correlated demand caused by cross-selling. We furthermore note that Park and Seo (2013) proposed an inventory model considering purchase dependence, which is similar to cross-selling; however, they did not consider backordering.

Zhang et al. (2012) asserted that the problem was essentially identical to the NP-hard JRP model; thus, they developed a heuristic algorithm for the near-optimal solution. However, the problem is in fact similar to the

* Corresponding author. 8 Clarkson Ave, Potsdam, NY 13676, USA. .
E-mail address: cxiang@clarkson.edu (C. Xiang).

https://doi.org/10.1016/j.ijpe.2018.03.007
Received 23 August 2017; Received in revised form 7 February 2018; Accepted 7 March 2018
Available online 8 March 2018
0925-5273/

JRP with a strict-cyclic policy (JRPSC), and it has been proven that the JRPSC has polynomial algorithms (Viswanathan, 1996; Wildeman et al., 1997; Lee and Yao, 2003). In this paper, we demonstrate that the inventory model proposed by Zhang et al. (2012), which considers demand association caused by cross-selling, holds similar mathematical properties. It follows that an exact algorithm with polynomial complexity should exist for determining the global optima.

The remainder of the paper is organized as follows. In section 2, we briefly introduce the inventory model and heuristic algorithm proposed by Zhang et al. (2012). Section 3 provides the properties of the inventory cost function, based on which the exact algorithm with polynomial complexity is presented. Section 4 discusses a numerical example for illustration, and reports on the performance of the proposed algorithm for solving larger-scale problems. The paper is concluded in section 5.

2. Model and heuristic algorithm

Consider a multi-item inventory system consisting of one major item and n minor items. The demand rate for the major item, denoted by D, is independent; however, demand for minor item i $(i = 1, \ldots, n)$ will be increased by major item sales as a result of the cross-selling effect. Conversely, if the major item is stocked out, the demand for minor item decreases to a certain degree. Denote by D_i the demand rate for minor item i without stock-outs of the major item, and by λ_i the loss rate; that is, the lost sales of i caused by one unit shortage of the major item. Then, the demand rate of minor item i will be decreased to $D_i - \lambda_i D$ (≥ 0) when the major item is stocked out.

Suppose that the unmet demand for the major item is completely backordered because of monopolization, while the minor item demand must be met without stock-outs for the sake of management convenience. Let C_b denote the cost of maintaining one major item unit backordered for one unit time. Moreover, let A and A_i denote the ordering cost of the major item and minor item i, respectively, where C_h and C_{hi} are the corresponding inventory holding costs. Let T and F denote the order cycle and fill rate (percentage of demand satisfied from the shelf stock) of the major item, respectively. Based on the strict-cyclic policy whereby the minor item order cycle is a positive integer multiple (denoted by k_i) of the major item order cycle, the total cost (per unit time) function is formulated as:

$$\Gamma(\mathbf{K}, T, F) = \frac{A + \sum_{i=1}^{n} A_i/k_i}{T} + \frac{C_b + C_h'}{2} DTF^2 - C_b DTF + \frac{C_b - \sum_{i=1}^{n} \lambda_i C_{hi}}{2} DT + \frac{\sum_{i=1}^{n} C_{hi} D_i k_i}{2} T \tag{1}$$

where $C_h' = C_h + \sum_{i=1}^{n} \lambda_i C_{hi}$ and $\mathbf{K} = \{k_1, \ldots, k_n\}$.

For a given $\mathbf{K}$, the optimal cost of $\Gamma(\mathbf{K}, T)$ is $\Gamma(\mathbf{K}) = 2\sqrt{G_0(G_3 - G_2^2/(4G_1))}$, where $G_0 = A + \sum A_i/k_i$, $G_1 = (C_b + C_h')D/2$, $G_2 = C_b D$, and $G_3 = [G_2 + \sum(k_i D_i - \lambda_i C_{hi} D)]/2$. Zhang et al. (2012) developed a heuristic algorithm by improving Nilsson's method (Nilsson et al., 2007) in order to minimize function $\Gamma(\mathbf{K})$. The heuristic iteratively selects $\mathbf{K}$ to balance the ordering and inventory holding costs as far as possible, until no further improvement can be made on the total cost. To this end, the algorithm introduces the ratio of the ordering cost to the sum of the inventory holding and backordering costs for each item, and adjusts the value of k_i, causing the ratio to be as close as possible to 1.

Zhang et al. (2012) reported that at least 70% of small-scale numerical examples (where each includes no more than 25 minor items) in their computational experiment can reach the optima by means of the heuristic, while the maximum error is no larger than 2.89% if threshold values of the ratio are appropriately selected (for their numerical examples, 0.6 and 0.4 were suggested as the thresholds for the low and high major ordering costs, respectively). They furthermore demonstrated that the heuristic is thousands of times faster than the branch-and-bound (BB) algorithm provided by Lingo 9.0 (programming software). Overall, Zhang's heuristic for their JRP model performs rather effectively.

However, the heuristic performance is affected by the threshold of the ratio, which must be set in advance, and for which they did not provide effective methods for determining an appropriate value. According to the computational experiment, the maximum error could even reach 4.72% if the ratio threshold was not appropriately selected (Zhang et al., 2012). Moreover, as the heuristic algorithm does not guarantee global optimality, we cannot evaluate its performance for larger-scale problems.

3. Exact algorithm

3.1. Model transformation

It is easy to establish that $\Gamma(\mathbf{K}, T, F)$ is convex in F. Thus, the first-order optimality condition with respect to F yields the optimal value of F, as follows:

$$F^* = \frac{C_b}{C_b + C_h + \sum_{i=1}^{n} \lambda_i C_{hi}} = \frac{C_b}{C_b + C_h'} \tag{2}$$

Substituting F^* into $\Gamma(\mathbf{K}, T, F)$ and rearranging terms, the optimal cost function is recast as

$$\Gamma(\mathbf{K}, T) = \frac{A}{T} + \frac{hT}{2} + \frac{1}{T}\sum_{i=1}^{n} A_i/k_i + \frac{T}{2}\sum_{i=1}^{n} h_i k_i \tag{3}$$

where $h = (F^* C_h' - \sum_{i=1}^{n} \lambda_i C_{hi})D$ and $h_i = C_{hi} D_i$. It can easily be proven that $\sum h_i > |h|$ as a result of $D_i - \lambda_i D \geq 0$, which guarantees that the optimal value of T is finite and the optima of function $\Gamma(\mathbf{K}, T)$ exist.

It appears that Eq. (3) has a similar formulation to the traditional JRPSC problem, the global optima of which may be determined by Goyal's enumeration algorithm (Goyal, 1974; Khouja and Goyal, 2008). However, because of the additional term $\frac{hT}{2}$ compared to the standard JRPSC model, Goyal's algorithm cannot in fact be directly applied to our "*quasi*" JRP problem (see computational results in section 4). In the following section, we develop an exact algorithm that can determine the global optima.

3.2. Properties of inventory cost function

Denote $\overline{\Gamma}(\mathbf{K}, T) = \frac{1}{T}\sum_{i=1}^{n} A_i/k_i + \frac{T}{2}\sum_{i=1}^{n} h_i k_i$, and define a new function as

$$\overline{\Gamma}(T) = \min_{\mathbf{K}} \overline{\Gamma}(\mathbf{K}, T)$$

Optimizing function $\Gamma(\mathbf{K}, T)$ is equivalent to optimizing function (4), as follows:

$$\Gamma(T) = \frac{A}{T} + \frac{hT}{2} + \overline{\Gamma}(T) = \frac{A}{T} + \frac{hT}{2} + \min_{\mathbf{K}} \overline{\Gamma}(\mathbf{K}, T) \tag{4}$$

It has been proven that, as a function in the form of Eq. (4), $\Gamma(T)$ exhibits the following properties (Viswanathan, 1996; Wildeman et al., 1997; Lee and Yao, 2003).

Property 1. $\Gamma(T)$ is a piecewise convex function and all the junction points are determined by

$$t_i(k_i) = \sqrt{\frac{2A_i}{h_i}}\sqrt{\frac{1}{k_i} - \frac{1}{k_i + 1}}, \quad k_i = 1, 2, \ldots; \quad i = 1, \ldots, n \tag{5}$$

Eq. (5) indicates that the junction points decrease in k_i; therefore, we have $t_i(1) > t_i(2) > \ldots > t_i(k_i) > t_i(k_i + 1) > \ldots$ $(i = 1, \ldots, n)$.

Property 2. With a given T, the optimal value of k_i is determined by

$$k_i(T) = \begin{cases} 1, & \text{if } t_i(1) \leq T \\ k_i, & \text{if } t_i(k_i) \leq T \leq t_i(k_i - 1), k_i \geq 2 \end{cases}, \quad i = 1, \ldots, n, \tag{6}$$

which is equivalent to $k_i(T) = -\frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{8A_i}{h_i T^2}}$. Here, X represents the smallest integer that is not smaller than X. Therefore, if T is located between two adjacent junction points of function $\Gamma(T)$, namely $t_i(k_i) < T \le t_i(k_i - 1)$, the corresponding optimal value of multiple k_i can be determined by Eq. (6), so that the optimum may easily be obtained. Because the intervals formed by adjacent junction points contain all possible values of T, searching all of these intervals will yield the global optimal solution.

Moreover, in order to complete the search procedure within finite steps, we need to determine the bounds of the optimal order cycle T^*. Considering Eq. (3) for a given $\mathbf{K}$, the optimal value of order cycle T is

$$T(\mathbf{K}) = \sqrt{\frac{2\left(A + \sum_{i=1}^{n} A_i/k_i\right)}{h + \sum_{i=1}^{n} h_i k_i}} \tag{7}$$

When $k_i = 1$ for all i, $T(\mathbf{K})$ takes its maximum value, implying that

$$T^* \le \sqrt{2\left(A + \sum_{i=1}^{n} A_i\right) \Big/ \left(h + \sum_{i=1}^{n} h_i\right)}$$

For all $k_i = 1$, $(i = 1, \ldots, n)$, the best inventory cost is $\overline{C} = \sqrt{2(A + \sum_{i=1}^{n} A_i)(h + \sum_{i=1}^{n} h_i)}$. Obviously, the optimal inventory cost should not be larger than $\overline{C}$; thus, we have

$$\Gamma\left(\mathbf{K}, T^*\right) = \frac{A}{T^*} + \frac{1}{T^*}\sum_{i=1}^{n} A_i/k_i + \frac{T^*}{2}\left(h + \sum_{i=1}^{n} h_i k_i\right) \le \overline{C}$$

All terms in the above inequality are positive, and it follows that $A/T^* \le \overline{C}$; that is,

$$T^* \ge A/\overline{C} = A \Big/ \sqrt{2\left(A + \sum_{i=1}^{n} A_i\right)\left(h + \sum_{i=1}^{n} h_i\right)}$$

Hence, for the optimal order cycle T^*, the upper bound U and lower bound L can be determined, respectively, as

$$U = \sqrt{2\left(A + \sum_{i=1}^{n} A_i\right) \Big/ \left(h + \sum_{i=1}^{n} h_i\right)} \tag{8}$$

$$L = A \Big/ \sqrt{2\left(A + \sum_{i=1}^{n} A_i\right)\left(h + \sum_{i=1}^{n} h_i\right)} \tag{9}$$

where $T^* \in [L, U]$.

It should be noted that Eq. (9) provides an initial lower bound for the optimal order cycle T^*. For an iterative algorithm, we can dynamically update the lower bound as follows:

$$T^* \ge A/\widehat{C} \tag{10}$$

where $\widehat{C}$ is the lowest cost obtained thus far.

3.3. Algorithm

$\Gamma(T)$ provides a piecewise convex function, so that we can determine the optimal value piece by piece. Specifically, we firstly calculate all the junction points within L and U using Eq. (5). Then, we sort all these points together and the interval of $[L, U]$ is split into multiple sub-intervals by each pair of sorted adjacent junction points. For each sub-interval, function $\Gamma(T)$ is smoothly convex, and the optimal value of the multiples $\mathbf{K} = \{k_1, \ldots, k_n\}$ can be determined by Eq. (6), so we can easily obtain the local optima for each sub-interval. Among all of these local optima, that with the minimal inventory cost provides the global optimal solution.

The above procedure is depicted in Fig. 1, and detailed as follows.

Step 1. Determine the optimal fill rate as $F^* = \frac{C_b}{C_b + C_h'}$. Let $h = (F^* C_h' - \sum_{i=1}^{n} \lambda_i C_{hi})D$ and $h_i = C_{hi} D_i$ $(i = 1, \ldots, n)$. Calculate the bounds of the optimal order cycle T^* using Eqs. (8) and (9), yielding the upper bound U and lower bound L, respectively.
Step 2. For minor item i $(i = 1, \ldots, n)$, exhaust all $k_i \in [l_i, u_i]$ in order to determine all junction points $t_i(k_i)$ on interval $[L, U]$ using Eq. (5), where $U \ge t_i(l_i) > \ldots > t_i(u_i) \ge L$. As a result, n groups of junction points are yielded for all minor items: $\{t_1(l_1), \ldots, t_1(u_1)\}$, …, $\{t_i(l_i), \ldots, t_i(u_i)\}$, …, $\{t_n(l_n), \ldots, t_n(u_n)\}$.
Step 3. Sort all the junction points of n minor items in increasing order, and if multiple junction points are identical, merge these. Denote by $\{\tau_0, \tau_1, \tau_2, \ldots, \tau_j, \tau_{j+1}, \ldots, \tau_S, \tau_{S+1}\}$ all the increasingly sorted junction points, where $\tau_0 = L$ and $\tau_{S+1} = U$.
Step 4. On each sub-interval $[\tau_j, \tau_{j+1}]$ $(j = 0, 1, \ldots, S)$, determine the optimal value of $\mathbf{K}$ as $\widehat{\mathbf{K}}_j = \{\widehat{k}_1, \ldots, \widehat{k}_n\}$ from Eq. (6). Moreover, substituting $\widehat{\mathbf{K}}_j$ into Eq. (7) yields the corresponding optimal order cycle $T(\widehat{\mathbf{K}}_j)$ on sub-interval $[\tau_j, \tau_{j+1}]$.
Make judgment: Considering the convexity of function $\Gamma(T)$, if $T(\widehat{\mathbf{K}}_j) \in [\tau_j, \tau_{j+1}]$, the optimal order cycle on sub-interval $[\tau_j, \tau_{j+1}]$ is $\widehat{T}_j = T(\widehat{\mathbf{K}}_j)$; if $T(\widehat{\mathbf{K}}_j) \le \tau_j$, $\widehat{T}_j = \tau_j$; otherwise, $\widehat{T}_j = \tau_{j+1}$. Substituting $\widehat{T}_j$ and $\widehat{\mathbf{K}}_j$ into Eq. (3) yields the local optimal inventory cost on sub-interval $[\tau_j, \tau_{j+1}]$ $(j = 0, \ldots, S)$, which is denoted by $\widehat{\Gamma}_j$.
Step 5. Let $\widehat{\Gamma}_r = \min\{\widehat{\Gamma}_0, \widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_j, \ldots, \widehat{\Gamma}_S\}$; then, $\widehat{\Gamma}_r$ is the global optimal inventory cost, and the corresponding order cycle $\widehat{T}_r$ and multiples $\widehat{\mathbf{K}}_r$ are the global optimal solution.

3.4. Computational complexity

Eq. (5) demonstrates that when $t_i(k_i)$ is sufficiently small, $k_i(T)$ will be sufficiently large and linearly proportional to $1/t_i(k_i)$. Furthermore, the minimum value of lower bound L is bounded by $O(1/n)$ (see Eq. (9)), as with $t_i(k_i)$. Thus, the maximum value of $k_i(T)$, that is, the maximum number of junction points for item i, is bounded by $O(n)$. It follows that for all minor items, at most $O(n^2)$ junction points will be obtained. It should be noted that the $O(n)$ junction points of each item have been sorted when deriving from Eq. (5). The further sorting of n groups of $O(n)$ sorted points carries a computational complexity of $O(n^2 \log n)$, based on the heap structure (Chowdhury and Kaykobad, 2001).

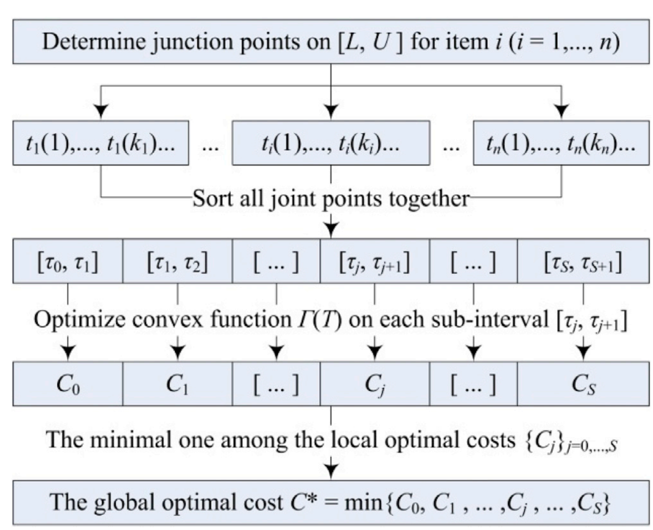

Fig. 1. Procedure of exact algorithm.

Moreover, the algorithm must check all sub-intervals for n items; thus, in step 4, in total we check at most $O(n \times n) = O(n^2)$ sub-intervals. For each sub-interval, the algorithm calculates the optimal value of k_i for each item, which requires a total of n computation times. Obviously, in order to check all sub-intervals, the computational effort is bounded by $O(n \times n^2) = O(n^3)$. With the sorting manipulation, the total computational effort of the above algorithm is bounded by $O(n^2 \log n) + O(n^3)$, implying that the proposed algorithm is polynomial.

4. Numerical computations

4.1. Illustrative numerical example

Consider an inventory system consisting of one major item and three minor items, the parameters of which are listed in Table 1. We use the proposed algorithm to solve the problem.

Step 1. The optimal fill rate is $F^* = \frac{C_b}{C_b + C_h} = 0.7667$; the values of h and h_i $(i = 1, \ldots, 3)$ are $h = (F^* C_h' - \sum_{i=1}^{n} \lambda_i C_{hi})D = 214.7692$, $h_1 = C_{h1} D_1 = 85.5$, $h_2 = C_{h2} D_2 = 16.92$, and $h_3 = C_{h3} D_3 = 4.2$. Calculate the upper and lower bounds of the optimal order cycle T^* using Eqs. (8) and (9), respectively, yielding $U = 1.2962$ and $L = 0.2400$.
Step 2. Determine junction points on interval $[0.2400, 1.2962]$ for all minor items using Eq. (5), which are listed in Table 2.

We observe that for item 2, when $k_2 < 2$ or $k_2 > 11$, the junction point jumps out of interval $[L, U] = [0.2400, 1.2962]$. Thus, we need only select the junction points of item 2 for $2 \le k_2 \le 11$. Similarly, we consider junction points with $1 \le k_1 \le 5$ for item 1, and $2 \le k_3 \le 12$ for item 3.

Step 3. Sort the end points of interval $[L, U]$ and the 26 junction points selected above in increasing order, which yields 27 sub-intervals: $I_0 = [0.2400,\ 0.2471]$, $I_1 = [0.2471,\ 0.2498]$, …, $I_{26} = [1.2599,\ 1.2962]$ (see Table 3).
Step 4. On each sub-interval, determine the optimal value of $\mathbf{K}$. On the first sub-interval $I_0 = [0.2400, 0.2471]$, for item 1, we have (see Table 2):

$$I_0 = [0.2400,\ 0.2471] \subseteq [0.2111,\ 0.2498] = [t_1(6),\ t_1(5)]$$

According to Eq. (6), the optimal value of k_1 on sub-interval $I_0 = [0.2400, 0.2471]$ is $\widehat{k}_1 = 6$; similarly, we have $\widehat{k}_2 = 12$ and $\widehat{k}_3 = 13$. Substituting $\widehat{\mathbf{K}}_0 = (\widehat{k}_1, \widehat{k}_2, \widehat{k}_3) = (6, 12, 13)$ into Eq. (7) yields $T(\widehat{\mathbf{K}}_0) = 0.4950$ for sub-interval $[0.2400, 0.2471]$.
Make judgment: As the extreme point $T(\widehat{\mathbf{K}}_0) = 0.4950 > 0.2471$, considering the convexity of function $\Gamma(T)$, the local optimum on the sub-interval should be the upper end point, namely $\widehat{T}_0 = 0.2471$. Substituting $\widehat{T}_0$ and $\widehat{\mathbf{K}}_0$ into Eq. (3) yields the optimal inventory cost $\widehat{\Gamma}_0 = 610.26$ on the first sub-interval $I_0 = [0.2400, 0.2471]$.
By the same manipulation, we can obtain all $T(\widehat{\mathbf{K}}_j)$ for each sub-interval, as well as the corresponding local optima $\widehat{T}_j$ and $\widehat{\Gamma}_j$.

Step 5. The global optimal inventory cost is $\Gamma^* = \min\{\widehat{\Gamma}_0, \widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_{26}\} = \widehat{\Gamma}_{22} = 390.67$, while the optimal solution is $F^* = 0.7667$, $\mathbf{K}^* = \widehat{\mathbf{K}}_{22} = \{2, 3, 4\}$, and $T^* = \widehat{T}_{22} = 0.8618$.

The shape of function $\Gamma(T)$ for the illustrative numerical example, which is a piecewise convex function, is depicted in Fig. 2.

4.2. Application of enumeration algorithm

As mentioned previously, Goyal's enumeration algorithm (GE algorithm for short) is not applicable to our problem, because of the term $\frac{hT}{2}$ in the objective function. However, we can equally recast the problem as

$$\begin{cases} \min \Gamma(\mathbf{K}, T) = \dfrac{A}{T} + \dfrac{1}{T}\left(\displaystyle\sum_{i=1}^{n} A_i/k_i + A_0/k_0\right) + \dfrac{T}{2}\left(\displaystyle\sum_{i=1}^{n} h_i k_i + h k_0\right) \\ \text{s.t.} \qquad k_0 = 1 \end{cases}, \quad (11)$$

where $A_0 \to 0$ is a sufficiently small positive number, so that the problem is transformed into a JRPSC model with $n+1$ items. We can adapt the GE algorithm to solve the equivalent model (11) by setting k_0 to 1.
For the illustrative example above, the smallest cost yielded by the adapted GE algorithm is $\Gamma = 392.59$, which is inferior to that of the exact algorithm, implying that the GE algorithm cannot be directly applied to the JRP model in this study. Further numerical computations also indicate that the adapted GE algorithm cannot determine the optimal solution based on model (11), as reported below.

4.3. Application of algorithm to larger-scale problems

In order to evaluate the computational performance of the algorithm when solving larger-scale problems, we construct 100 numerical examples by randomly selecting the parameters $D \in [100,\ 10000]$, $A \in [10,\ 1000]$, $C_h \in [0.1,\ 10]$, $C_b \in [1,\ 100]$, $\lambda_i \in [0.05,\ 1]$, $D_i \in [\lambda_i D,\ 5D]$, $A_i \in [10,\ 100]$, and $C_{hi} \in [0.001,\ 5]$, each of which includes 25 minor items ($n = 25$). Under the same scheme, for $n = 50,\ 100,\ 150,\ 200,\ 300,\ 400,\ 500,\ 600,\ 700,\ 800,\ 900,\ 1000$, we also respectively generate 100 numerical examples, and in total we test 13 groups of data (where each group includes 100 examples).
All examples are solved by the adapted GE and exact algorithm, on a computer with a Duo processor with Core i7-7700 CPU, 3.60 GHz, and 32.00 G memory. Table 4 displays the average and maximum computational times (Avg. time and Max. time) of the two algorithms, as well as the average and maximum relative deviations from the optima (Avg. dev. and Max. dev.) of the adapted GE algorithm.
Table 4 demonstrates that the adapted GE algorithm cannot obtain the optimal solution. For various examples, the relative deviation of the total cost of the adapted GE algorithm reaches more than 10%, compared with the exact algorithm. It appears that the adapted GE algorithm based

Table 1
Numerical example for illustration.

Parameters	Major item	Minor items		
		1	2	3
Ordering cost (A, A_i)	100	80	70	20
Holding cost (C_h, C_{hi})	0.5	0.095	0.0235	0.01
Demand rate (D, D_i)	600	900	720	420
Loss rate (λ_i)	–	1.0	0.5	0.2
Backordering cost (C_b)	2	–	–	–

Table 2
Junction points of all minor items.

k_i	1	2	3	4	5	6	7	8	9	10	11	12	13
$t_1(k_1)$	**0.9673**	**0.5585**	**0.3949**	**0.3059**	**0.2498**	0.2111	…						
$t_2(k_2)$	2.0340	**1.1743**	**0.8304**	**0.6432**	**0.5252**	**0.4439**	**0.3844**	**0.3390**	**0.3032**	**0.2743**	**0.2504**	0.2303	…
$t_3(k_3)$	2.1822	**1.2599**	**0.8909**	**0.6901**	**0.5634**	**0.4762**	**0.4124**	**0.3637**	**0.3253**	**0.2942**	**0.2686**	**0.2471**	0.2288

The bold faced numbers are the junction points with the given k.

Table 3
Optimal solution on each sub-interval.

Sub-intervals														
τ_j	0.2400	0.2471	0.2498	0.2504	0.2686	0.2743	0.2942	0.3032	0.3059	0.3253	0.3390	0.3637	0.3844	0.3949
τ_{j+1}	0.2471	0.2498	0.2504	0.2686	0.2743	0.2942	0.3032	0.3059	0.3253	0.3390	0.3637	0.3844	0.3949	0.4124
$\widehat{k}_1$	6	6	5	5	5	5	5	5	4	4	4	4	4	3
$\widehat{k}_2$	12	12	12	11	11	10	10	9	9	9	8	8	7	7
$\widehat{k}_3$	13	12	12	12	11	11	10	10	10	9	9	8	8	8
$T(\widehat{\mathbf{K}}_j)$	0.4950	0.4963	0.5251	0.5313	0.5329	0.5395	0.5412	0.5484	0.5879	0.5900	0.5991	0.6014	0.6114	0.6682
$\widehat{T}_j$	0.2471	0.2498	0.2504	0.2686	0.2743	0.2942	0.3032	0.3059	0.3253	0.3390	0.3637	0.3844	0.3949	0.4124
$\widehat{\Gamma}_j$	610.26	606.34	605.40	579.78	572.72	550.38	541.65	539.13	521.11	510.07	492.85	480.93	475.48	465.98

Sub-intervals													
τ_j	0.4124	0.4439	0.4762	0.5252	0.5585	0.5634	0.6432	0.6901	**0.8304**	0.8909	0.9673	1.1743	1.2599
τ_{j+1}	0.4439	0.4762	0.5252	0.5585	0.5634	0.6432	0.6901	0.8304	**0.8909**	0.9673	1.1743	1.2599	1.2962
$\widehat{k}_1$	3	3	3	3	2	2	2	2	**2**	2	1	1	1
$\widehat{k}_2$	7	6	6	5	5	5	4	4	**3**	3	3	2	2
$\widehat{k}_3$	7	7	6	6	6	5	5	4	**4**	3	3	3	2
$T(\widehat{\mathbf{K}}_j)$	0.6714	0.6848	0.6883	0.7040	0.7968	0.8019	0.8251	0.8313	**0.8618**	0.8701	1.0747	1.1308	1.1462
$\widehat{T}_j$	0.4439	0.4762	0.5252	0.5585	0.5634	0.6432	0.6901	0.8304	**0.8618**	0.8909	1.0747	1.1743	1.2599
$\widehat{\Gamma}_j$	451.74	439.88	426.78	420.10	418.85	403.67	397.74	390.94	**<u>390.67</u>**	390.88	390.80	392.34	394.35

The bold faced column is the optimal solution with the optimal inventory cost (390.67) underlined.

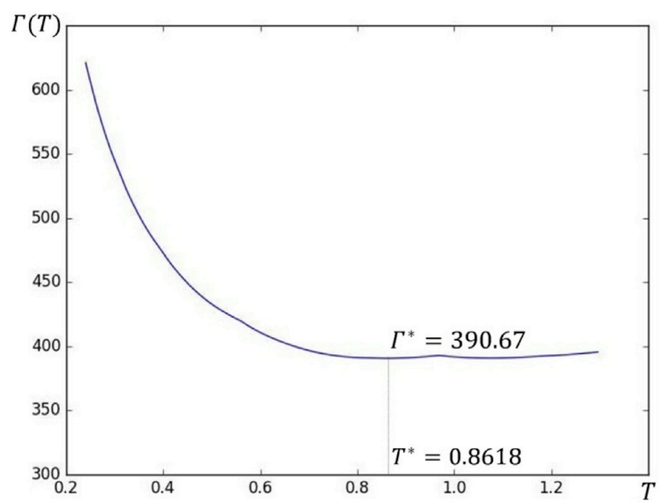

Fig. 2. Shape of piecewise convex function $\Gamma(T)$ of illustrative numerical example.

on model (11) always leads to inferior solutions. The explanation for this is that the absolute value of $h = (F^* C_h' - \sum_{i=1}^{n} \lambda_i C_{hi}) D$ may be significantly larger than $h_i = C_{hi} D_i$, so the term $\frac{hT}{2}$ plays an important role in the cost function and should not be ignored by setting k_0 as equal to 1. Specifically, the data in the Avg. dev. column of Table 4 indicate that with the adapted GE algorithm, the average deviation from the optima becomes larger for more minor items. The reason for this is that more minor items result in a larger absolute value of h, resulting in a greater impact by the term $\frac{hT}{2}$.

Moreover, the computational results demonstrate that the polynomial algorithm can efficiently obtain the exact optima for problems with up to 1000 minor items, which will provide a highly competent solution in practical applications.

5. Conclusion

This paper presents an exact algorithm for the problem of JRP with complete backordering and correlated demand, as studied by Zhang et al. (2012). We prove that the proposed algorithm is of polynomial complexity and thus the problem is not NP-hard, which amends the previous study. The computational results demonstrate that the proposed polynomial algorithm is highly efficient for large-scale problems, which offers a competent solution for practical applications.

It should also be mentioned that our algorithm can be applied to solving inventory models related to JRPSC. Moreover, the trick of determining the optimal solution based on piecewise convexity or concavity can be used in various other contexts, such as those of Zhou et al. (2013, 2017) and Li et al. (2017).

Table 4
Computational results.

n	Adapted GE				Exact algorithm	
	Avg. time (s)	Max. time (s)	Avg. dev.	Max. dev.	Avg. time (s)	Max. time (s)
25	0.005	0.02	4.47%	13.27%	**0.009**	**0.05**
50	0.02	0.05	6.33%	12.33%	**0.03**	**0.11**
100	0.12	0.20	8.12%	13.85%	**0.14**	**0.69**
150	0.35	0.48	9.04%	13.63%	**0.28**	**1.05**
200	0.77	1.08	9.59%	13.29%	**0.52**	**1.75**
300	2.41	3.05	10.07%	12.39%	**1.07**	**3.20**
400	5.63	7.09	10.33%	12.42%	**2.05**	**9.31**
500	10.87	13.30	10.64%	12.49%	**3.15**	**8.91**
600	18.63	23.28	10.86%	13.07%	**4.47**	**8.75**
700	29.79	35.70	10.85%	12.31%	**5.98**	**11.77**
800	44.92	53.36	10.96%	12.44%	**7.87**	**14.58**
900	62.85	75.59	10.94%	12.54%	**10.14**	**17.14**
1000	86.88	102.69	11.16%	12.69%	**12.41**	**17.88**

Acknowledgements

The research is supported by the National Natural Science Foundation of China, under Grant Nos. 71271010 and 71571006, and the Aeronautical Science Foundation of China, under Grant No. 2014ZG51077. We extend our sincere thanks to the anonymous reviewers for their valuable suggestions and comments.

References

Ai, X.-Y., Zhang, J.-L., Wang, L., 2017. Optimal joint replenishment policy for multiple non-instantaneous deteriorating items. Int. J. Prod. Res. 55 (16), 4625–4642.

Braglia, M., Castellano, D., Gallo, M., 2016a. An extension of the stochastic Joint-Replenishment Problem under the class of cyclic policies. Operations Res. Lett. 44 (2), 278–284.

Braglia, M., Castellano, D., Frosolini, M., 2016b. Joint-replenishment problem under stochastic demands with backorders-lost sales mixtures, controllable lead times, and

investment to reduce the major ordering cost. J. Operational Res. Soc. 67 (8), 1108–1120.
Braglia, M., Castellano, D., Song, D., 2017. Distribution-free approach for stochastic Joint-Replenishment Problem with backorders-lost sales mixtures, and controllable major ordering cost and lead times. Comput. Operational Res. 79 (3), 161–173.
Chowdhury, R.A., Kaykobad, M., 2001. Sorting using heap structure. Int. J. Comput. Math. 77 (3), 347–354.
Cunha, J.O., Konstantaras, I., Melo, R.A., Sifaleras, A., 2017. On multi-item economic lot-sizing with remanufacturing and uncapacitated production. Appl. Math. Model. 50 (10), 772–780.
Goyal, S., 1974. Determination of optimum packaging frequency of items jointly replenished. Manag. Sci. 21 (4), 436–443.
Hashimoto, K., Matsubayashi, N., 2014. A note on dynamic monopoly pricing under consumption externalities. Econ. Lett. 124 (1), 1–8.
Huang, D., Zhou, H., Zhao, Q.-H., 2011. A competitive multi-product newsboy problem with partial product substitution. Omega 39 (3), 302–312.
Khouja, M., Goyal, S., 2008. A review of the joint replenishment problem literature: 1989–2005. Eur. J. Operational Res. 186 (1), 1–16.
Kouki, C., Babai, M.Z., Jemai, Z., Minner, S., 2016. A coordinated multi-item inventory system for perishables with random lifetime. Int. J. Prod. Econ. 181 (Part A), 226–237.
Lee, C.-Y., Lee, D., 2018. An efficient method for solving a correlated multi-item inventory system. Operations Res. Perspect. 5, 13–21.
Lee, F.-C., Yao, M.-J., 2003. A global optimum search algorithm for the joint replenishment problem under power-of-two policy. Comput. Operations Res. 30 (9), 1319–1333.
Li, S.-R., Zhang, R.-Q., Fang, W.-G., 2017. The newsvendor model with non-zero reference point based on cumulative prospect theory. Comput. Industrial Eng. 113 (11), 195–205.
Lippman, S.A., McCardle, K.F., 1997. The competitive newsboy. Operations Res. 45 (1), 54–65.
Netessine, S., Rudi, N., 2003. Centralized and competitive inventory models with demand substitution. Operations Res. 51 (2), 329–335.
Netessine, S., Zhang, F., 2005. Positive vs. negative externalities in inventory management: implications for supply chain design. Manuf. Serv. Operations Manag. 7 (1), 58–73.
Nilsson, A., Segerstedt, A., Van der Sluis, E., 2007. A new iterative heuristic to solve the joint replenishment problem using a spreadsheet technique. Int. J. Prod. Econ. 108 (1–2), 399–405.
Park, C., Seo, J., 2013. Consideration of purchase dependence in inventory management. Comput. Industrial Eng. 66 (2), 274–285.
Parlar, M., 1988. Game theoretic analysis of the substitutable product inventory problem with random demands. Nav. Res. Logist. 35 (3), 397–409.
Turnovsky, S.J., Monteiro, G., 2007. Consumption externalities, production externalities, and efficient capital accumulation under time non-separable preferences. Eur. Econ. Rev. 51 (2), 479–504.
Venkatachalam, S., Narayanan, A., 2016. Efficient formulation and heuristics for multi-item single source ordering problem with transportation cost. Int. J. Prod. Res. 54 (14), 4087–4103.
Viswanathan, S., 1996. A new optimal algorithm for the joint replenishment problem. J. Operational Res. Soc. 47 (7), 936–944.
Wildeman, R.E., Frenk, J.B.G., Dekker, R., 1997. An efficient optimal solution method for the joint replenishment problem. Eur. J. Operational Res. 99 (2), 433–444.
Zhang, R.-Q., Zhang, L.-K., Zhou, W.-H., et al., 2014. The multi-item newsvendor model with cross-selling and the solution when demand is jointly normally distributed. Eur. J. Operational Res. 236 (1), 147–159.
Zhang, R., Kaku, I., Xiao, Y., 2012. Model and heuristic algorithm of the joint replenishment problem with complete backordering and correlated demand. Int. J. Prod. Econ. 139 (1), 33–41.
Zhao, X., Atkins, D.R., 2008. Newsvendors under simultaneous price and inventory competition. Manuf. Serv. Operations Manag. 10 (3), 539–546.
Zhou, W., Wan, Q., Zhang, R.-Q., 2017. Choosing among hospitals in the subsidized health insurance system of China: a sequential game approach. Eur. J. Operational Res. 257 (2), 568–585.
Zhou, W.-H., Zhang, R.-Q., Zhou, Y.-W., 2013. A queueing model of supply chain with the form postponement strategy. Comput. Industrial Eng. 66 (4), 643–652.